



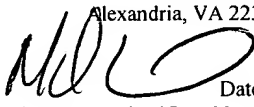
[40101/01101]

JPW
APF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s) : Mccombe et al.
Serial No. : 09/738,786
Filing Date : December 15, 2000
For : System and Method for Managing Client Processes
Group Art Unit : 2155
Examiner : Bharat Barot

Mail Stop: Appeal Brief-Patent
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

<p align="center">Certificate of Mailing</p> <p>I hereby certify that this correspondence is being deposited with U.S. Postal Services as first class mail in an envelope addressed to:</p> <p align="center">Mail Stop: Appeal Brief - Patents Commissioner for Patents P.O. Box 1450 Alexandria, VA 22313-1450</p> <p>By:  Date: September 30, 2005 Michael J. Marcin, (Reg. No. 48,198)</p>
--

TRANSMITTAL

In response to the Notice of Appeal filed August 4, 2005 and the Advisory Action dated August 22, 2005, transmitted herewith please find three copies of an Appeal Brief for filing in the above-identified application. Applicants hereby request a one-month extension. Please charge the Credit card of Fay Kaplun & Marcin, LLP in the amount of \$500.00. The Commissioner is hereby authorized to charge the **Deposit Account of Fay Kaplun & Marcin, LLP NO. 50-1492** for additional required fees. A copy of the paper is enclosed for that purpose.

Respectfully submitted,

Dated: September 30, 2005

By: 
Michael J. Marcin (Reg. No. 48,198)

Fay Kaplun & Marcin, LLP
150 Broadway, Suite 702
New York, NY 10038
(212) 619-6000 (telephone)
(212) 619-0276 (facsimile)

PATENT

Attorney Docket No.: 40101 - 01101

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:

Kevin McCombe et al.

Serial No.: 09/738,786

Filed: December 15, 2001

For: SYSTEM AND METHOD FOR
MANAGING CLIENT PROCESSES

Group Art Unit: 2155

Examiner: Bharat Barot

**Board of Patent Appeals and
Interferences**

Mail Stop: Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

In support of the Notice of Appeal filed August 4, 2005, and pursuant to 37 C.F.R.

§ 41.37, Appellants present in triplicate their appeal brief in the above-captioned application.

This is an appeal to the Board of Patent Appeals and Interferences from the Examiner's final rejection of claims 1-11 in the final Office Action dated May 19, 2005. The appealed claims are set forth in the attached Claims Appendix.

10/04/2005 BABRAHA1 00000056 09738786

01 FC:1402

500.00 0P

1. Real Party in Interest

This application is assigned to Wind River Systems, Inc., the real party in interest.

2. Related Appeals and Interferences

There are no other appeals or interferences which would directly affect, be directly affected, or have a bearing on the instant appeal.

3. Status of the Claims

Claims 1-11 have been rejected in the final Office Action. The final rejection of claims 1-11 is being appealed.

4. Status of Amendments

There have been no amendments submitted for this application by the Appellants.

5. Summary of Claimed Subject Matter

The present invention describes a system for managing client processes which includes a client task for executing the client processes and a manager task for queuing the client processes into the client task in priority order. (See Specification, p. 2, ll. 14-18). If a client process is not completed within a predetermined time period, the manager task will kill (or terminate) the client task. (See Id.). In an exemplary embodiment of the present invention, the system is implemented on a network 1 having a client-server relationship that includes network

segments 10-60, network buses 11-61, various network devices, and a server 15, wherein the server and the other devices are the clients. (See Id., p. 3, ll. 10-19). Since the server 15 may provide service to all other clients on the network 1, a processor 100 of the server 15 may be required to multitask in order to use the resources of the server 15 in the most efficient manner. (See Id., p. 3, ll. 19-22). While the network 1 is an exemplary embodiment on which the present invention may be implemented, it should be noted that the invention may be implemented on any computing device where multiple client processes or procedures may be run on a microprocessor. (See Id., p. 3, ll. 22-26).

As discussed above, the system manages the execution of the client process 170 within a client task 160. The manager task 150 of the system runs at a higher priority than the client task 160, whereby the execution of the manager task 150 will preempt the execution of the client task 160. (See Id., p. 5, ll. 1-4). The manager task 150 and client task 160 of the system may be considered processor management tools that work in conjunction to prevent the client process 170 from interfering with the execution of other client processes on the processor 100. (See Id., p. 5, ll. 4-6). The manager task 150 monitors the execution of the client process 170 within the client task 160 to ensure that the client process 160 does not deteriorate the availability of the processor 100. (See Id., p. 5, ll. 11-16). This deterioration may occur if the processor 100 is unable to complete the execution of the client process 170, thereby making the processor 100 unavailable to any other client processes waiting to be executed. (See Id., p. 5, ll. 6-9).

During the execution of the client process 170, the manager task 150 will expect a response from the client task 160 to indicate that the client process 170 is executing properly, wherein such an indication may be that the client process 170 is complete. (See Id., p. 5, ll. 19-23). If the manager task 150 has not received the indication within a predetermined period of time, the manager task 150 may restart the client task 160, thereby killing the execution of the client process 170 within the processor 100. (See Id., p. 5, ll. 23-26). Once the client task 160 has been restarted, the manager task 150 may then queue a new client process within the client task 160 to be executed by the processor 100. (See Id., p. 5, ll. 26-29). Therefore, the above system may be used to prevent the processor 100 from executing in a continuous processing loop, unable to complete the execution of an errant client process and unavailable to process other client processes.

6. Grounds of Rejection to be Reviewed on Appeal

- I. Whether claims 1-11 are unpatentable under 35 U.S.C. § 103(a) as obvious over U.S. Patent No. 6,470,346 to Morwood (hereinafter “the Morwood patent”) in view of U.S. Patent No. 6,769,019 to Ferguson. (hereinafter “the Ferguson patent”).

schedule of bandwidth priority is created, and when the connection between the client and the server is idle, the pre-selected hyperlinks are downloaded and stored in a cache on the user's hard drive. (See Id.).

B. The Cited Patents Do Not Disclose That the Manager Task Kills the Client Task When a Current One of the Client Processes is not Completed within a Predetermined Time Period as Recited in Claim 1

In the final Office Action, the Examiner acknowledged that the Morwood patent fails to disclose that "the manager task kills the client task when a current one of the client processes is not completed within a predetermined time period." (See 05/19/05 Office Action, p. 3, ¶ 5). However, the Examiner further stated that the Ferguson patent shows these claimed elements, thereby rendering the claimed subject matter obvious over the Morwood patent. (See Id.). Appellants respectfully disagree with the Examiner's rejection of claim 1.

The present application explains that the manager task will monitor a client process within the client task in order to ensure that the processor is continuously available. (See Id., p. 5, ll. 11-15). The recitation of claim 1 makes it clear that if a client process is not complete within a predetermined time period, the manager task may kill the execution of the client process within a processor by restarting the client task. If a client task with an errant client process is not killed, the processor may not complete the execution of the process, and the processor may end up in a continuous processing loop where the other client processes are waiting for the processor to become available. (See Id., p. 5, ll. 6-9). When the manager task

7. Grouping of Claims

Claims 1-11 may stand or fall together.

8. Argument

- I. The Rejection of Claims 1-11 Under 35 U.S.C. § 103(a) as Being Obvious Over U.S. Patent No. 6,470,346 to Morwood in view of U.S. Patent No. 6,769,019 to Ferguson Should Be Reversed.

A. The Examiner's Rejection

In the final Office Action, the Examiner rejected claims 1-11 Under 35 U.S.C. § 103(a) as being unpatentable over the Morwood patent in view of the Ferguson patent. (See 05/19/05 Office Action, p. 2, ¶ 4).

The Morwood patent describes a method for managing and performing computational tasks, wherein the method enables a requesting client to invoke a computation on a remote server. (See the Morwood patent, col. 1, ll. 28-30). This remote computation process allows the user to export any computationally intensive applications to a server that is appropriate for the execution of that particular application. (See Id., col. 1, ll. 50-63). The Ferguson patent describes a method for maximizing the use of available bandwidth while browsing the World Wide Web. (See the Ferguson patent, col. 2, l. 61 - col. 3, l. 20). Users may pre-select the Web pages they wish to view while viewing other content. (See Id.). A

kills a client task due to remaining incomplete for the predetermined time period, the manager task may then queue the next client process into the client task so that it may be executed by the processor. (See Id., p. 5, ll. 26-28). Therefore, the present invention will allow the resources of a processor in a server to remain continuously available to multitask for several clients despite the possibility of processing an invalid or improper client process.

The Examiner states that the Ferguson patent discloses killing a client task when a current one of the client processes is not completed within a predetermined time period. (See 05/19/05 Office Action, p. 3, ¶ 5). However, the Ferguson patent only discloses that a current time interval is compared to a preset threshold in order to determine if a Client Task Manager needs to issue a network activity. (See the Ferguson patent, col. 13, ll. 24-33). The network activity refers to initiating a download for updating the content of the Web server. (See Id., col. 13, ll. 56-59). Thus, the purpose of the preset threshold in the Ferguson patent is to determine if a download should be initiated, rather than if a client task should be killed as the Examiner contends. The preset threshold does not serve the same function as the predetermined time period of the present invention. The Ferguson patent goes on to state that any download or upload activity will be suspended if a request is made to use higher priority item, as to allow the higher priority item to have access to the connection. (See Id., col. 13, ll. 33-38 and col. 18, ll. 1-6). The suspended activity will only resume when all other higher priority requests have completed their data exchange. (See Id.). Therefore, a higher priority request will always take precedence over a lower priority network activity, regardless of how long the network activity

has been running. Therefore, the network activity has the potential to run indefinitely, so long as no higher priority requests are made. This potential result of the Ferguson patent is in contrast to the present invention that is designed to kill a request that is not executed within a predetermined time period and would, thus otherwise run indefinitely.

In addition, it is respectfully submitted that the Ferguson reference does not disclose killing a client task. The Examiner states that since ongoing network activity is suspended while a higher priority request is served, this implies that the network activity is killed. (See 05/19/05 Office Action, p. 3, ¶ 5). However, it is well known in the art that network activity can be suspended temporarily while another activity is serviced. Execution of the suspended activity can then be resumed from the point at which it was suspended. By suspending and resuming the activity, the Ferguson reference teaches away from the present invention. Killing a client task requires that execution of the task cannot be resumed. If the user desires to run the task after it is killed, the client task must be scheduled as a new task and executed starting at the beginning of the task. As stated previously, one purpose of killing the task is to prevent the task from running in a loop indefinitely. Thus, it would not be desirable to merely suspend the task, only to have the task resume running in the loop.

Thus, it is respectfully submitted that the Ferguson patent does not teach or suggest "that the manager task kills the client task when a current one of the client processes is not completed within a predetermined time period," as recited in claims 1.

Therefore, for at least these reasons, it is respectfully submitted that claim 1 is allowable. The Appellants respectfully request that the Board overturn the Examiner's rejection under 35 U.S.C. § 103(a) of claim 1. Because claims 2-5 depend from, and therefore include all the limitations of, claim 1, Appellants respectfully submit that these claims are allowable and request that the Examiner's rejections of these claims are also overturned.

The Examiner rejected claims 6-10 on the same grounds as claims 1-5, indicating that claims 6-10 were merely a method of operation for the apparatus of claims 1-5. The Examiner used the same rationale to reject claim 11, indicating that claim 11 was merely a computer-readable storage medium storing a set of instructions to manage the apparatus defined in claim 1. For the reasons stated above with respect to claim 1, Appellants respectfully submit that claims 6-11 are allowable and request that the Examiner's rejections of these claims are overturned.

8. Conclusions

For the reasons set forth above, Appellants respectfully request that the Board reverse the final rejections of the claims by the Examiner under 35 U.S.C. § 103(a), and indicate that claims 1-11 are allowable.

Respectfully submitted,

Date: September 30, 2005

By: 

Michael J. Marcin (Reg. No. 48,198)

Fay Kaplun & Marcin, LLP
150 Broadway, Suite 702
New York, NY 10038
Tel: (212) 619-6000
Fax: (212) 619-0276

CLAIMS APPENDIX

1. A system for managing a plurality of client processes, comprising:
 - a client task within which the client processes will be executed; and
 - a manager task running at a higher priority than the client task, the manager task queuing the client processes into the client task in priority order, wherein the manager task kills the client task when a current one of the client processes is not completed within a predetermined time period.
2. The system according to claim 1, wherein the manager task restarts the client task and queues a next one of the client processes into the client task.
3. The system according to claim 1, wherein the manager task restarts the client task and requeues the current client process into the client task.
4. The system according to claim 1, wherein the client task sends a response to the manager task indicating the execution of the current client process is complete.
5. The system according to claim 4, wherein the manager task, when receiving the response from the client task, queues a next one of the client processes into the client task.
6. A method for managing a plurality of client processes, comprising the steps of:

queuing a first one of the client processes into a client task, wherein the first client process is executed within the client task; and

killing execution of the client task by a manager task executing at a priority higher than that of the client task when the first client process is not completed within a predetermined time period.

7. The method according to claim 6, further comprising the step of:

releasing a first semaphore by the manager task, wherein the client task does not execute until the first semaphore is released by the manager task.

8. The method according to claim 7, further comprising the step of:

releasing a second semaphore by the client task indicating the execution of the first client process is complete.

9. The method according to claim 6, further comprising the steps of:

restarting the client task by the manager task; and

queuing a second one of the client processes into the client task.

10. The method according to claim 6, further comprising the steps of:

restarting the client task by the manager task; and

requeuing the first client process into the client task

11. A computer-readable storage medium storing a set of instructions, the set of instructions capable of being executed by a processor to manage a plurality of client processes, the set of instructions performing the steps of:

 queuing a first one of the client processes into a client task, wherein the first client process is executed within the client task; and

 killing execution of the client task by a manager task executing at a priority higher than that of the client task when the first client process is not completed within a predetermined time period.

Serial No.: 09/738,786
Group Art Unit: 2155
Attorney Docket No.: 40101 - 01101

EVIDENCE APPENDIX

No evidence has been entered or relied upon in the present appeal.

Serial No.: 09/738,786
Group Art Unit: 2155
Attorney Docket No.: 40101 - 01101

RELATED PROCEEDING APPENDIX

No decisions have been rendered regarding the present appeal or any proceedings related thereto.